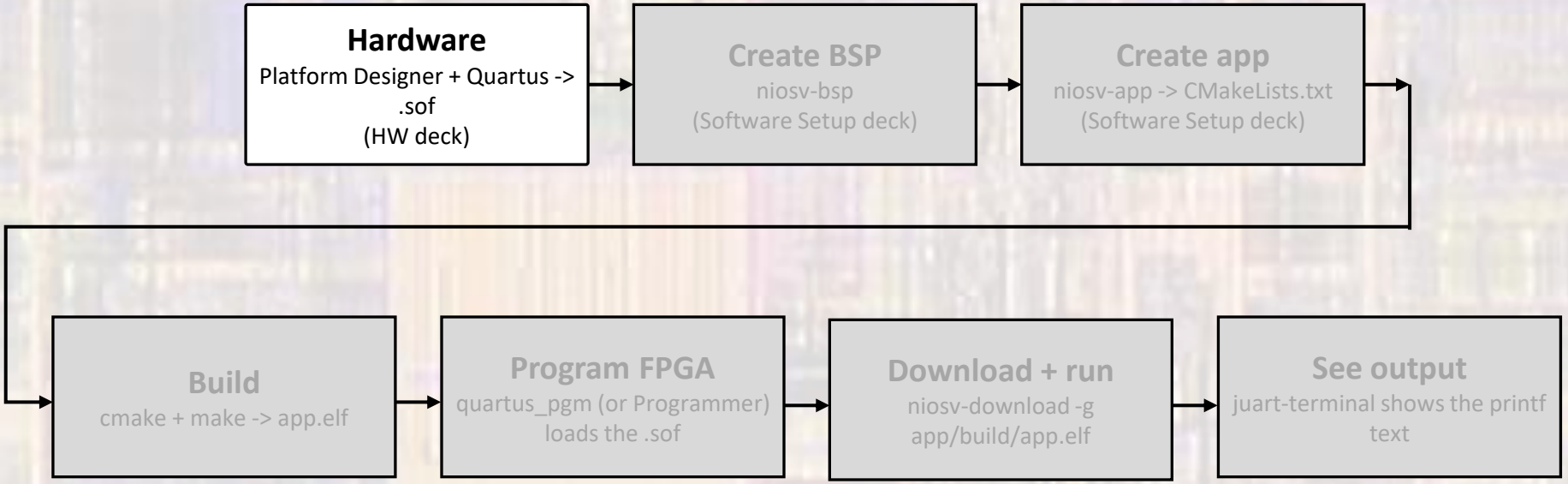
NIOS-V Hardware Setup

Last updated 10/29/26

NIOS-V – Hardware Setup

- From hardware to a running program



© tj

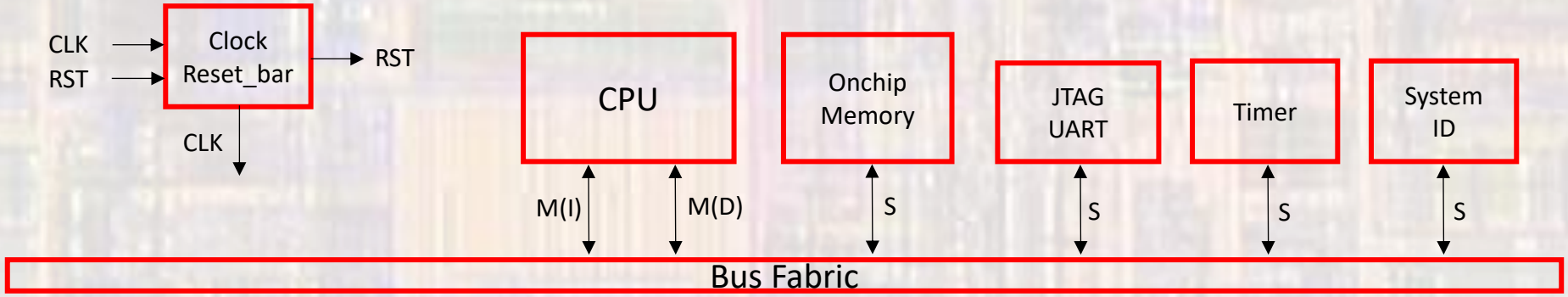
NIOS-V – Hardware Setup

- NIOS-V Embedded Design Suite
  - Configurable Processor
  - Selection of Peripherals
  - Board Support Package (BSP) for SW development

© tj

NIOS-V – Hardware Setup

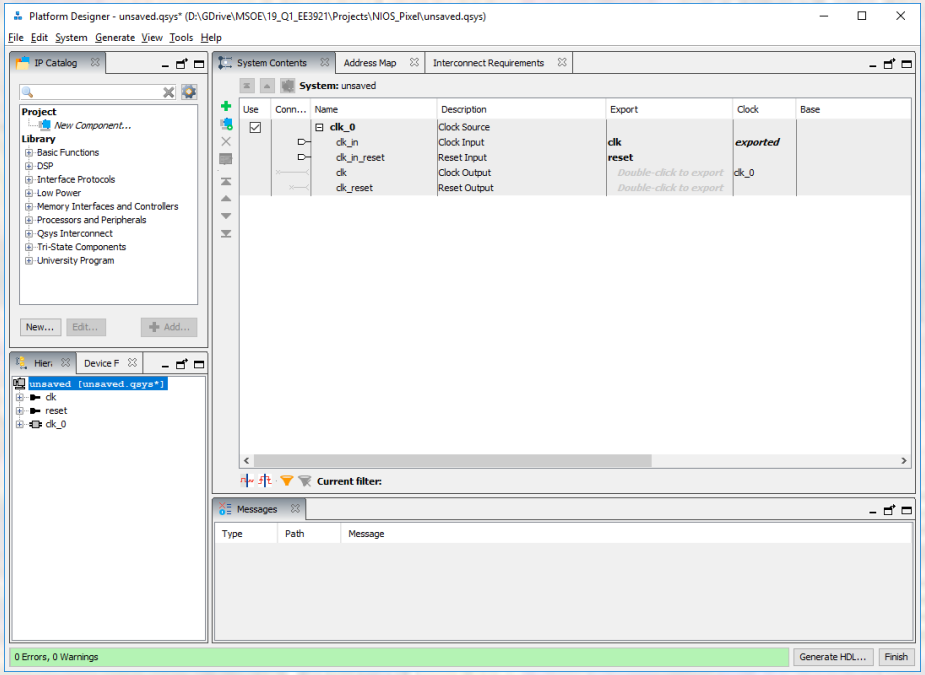
- Basic NIOS-V System
  - Create a processor system to allow printing to the console



© tj

NIOS-V – Hardware Setup

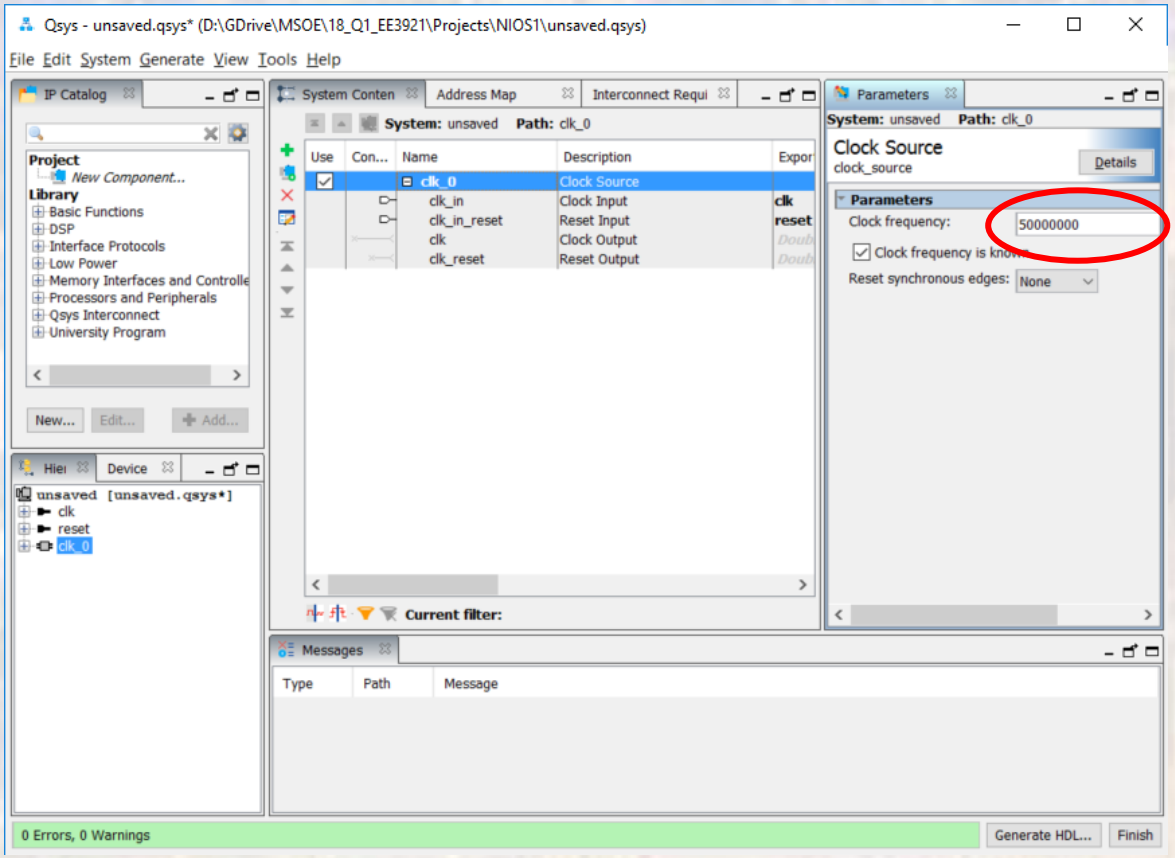
- Create a new Quartus project
  - Do not select a Simulation Tool in EDA Tool Settings
  - Be sure to follow all the steps in the Quartus Project Setup slides, NIOS-V requires all steps to be executed
- Open Tools → Platform Designer



© tj

NIOS-V – Hardware Setup

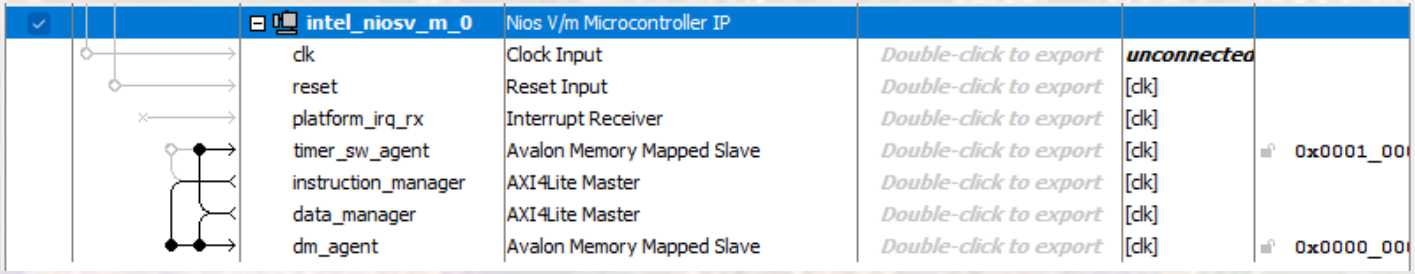
- Create NIOS-V System
  - Double Click on clk_0 - verify clk frequency = 50MHz



© tj

NIOS-V – Hardware Setup

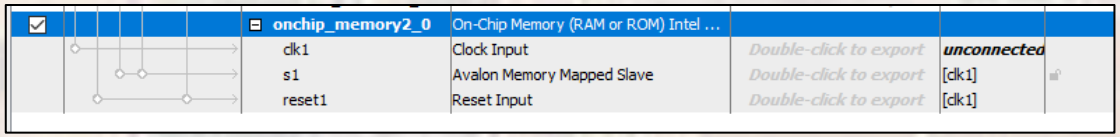
- Add NIOS-V
  - Processors and Peripherals → Embedded Processors → NIOS V/m Microcontroller
  - No other changes for now



- Add On-chip Memory
  - Basic Functions→ On Chip Memory → On Chip Memory (RAM or ROM)…

    RAM

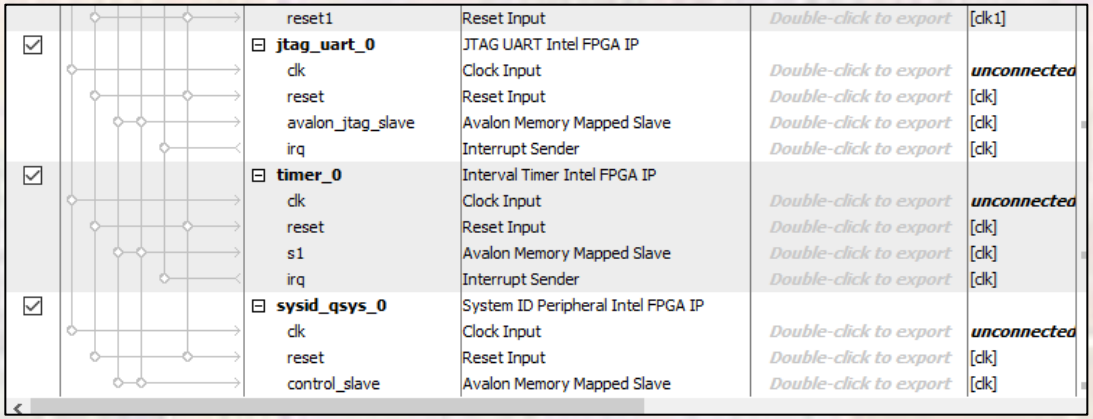
    Size = 100,000 bytes



© tj

NIOS-V – Hardware Setup

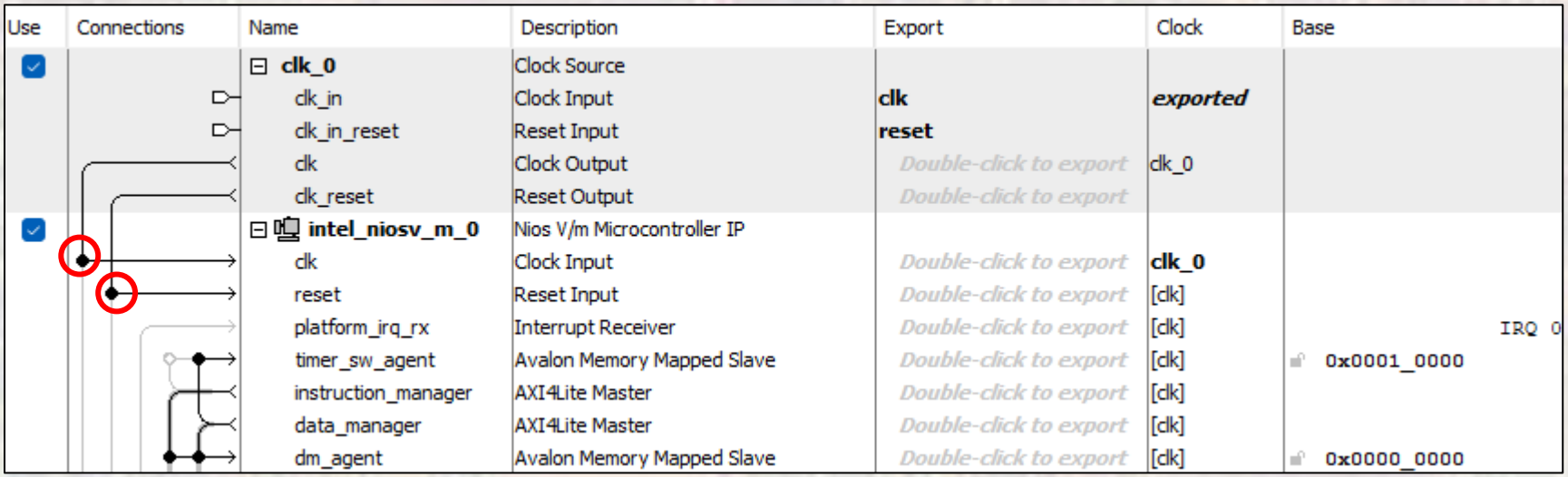
- Add JTAG
  - Interface Protocols → Serial→ JTAG Uart Intel FPGA IP
- Add Timer
  - Processors and Peripherals → Peripherals → Interval Timer Intel FPGA IP
- Add System ID
  - Basic Functions → Simulation; Debug and Verification → Debug and Performance → System ID Peripheral Intel FPGA IP



 © tj

NIOS-V – Hardware Setup

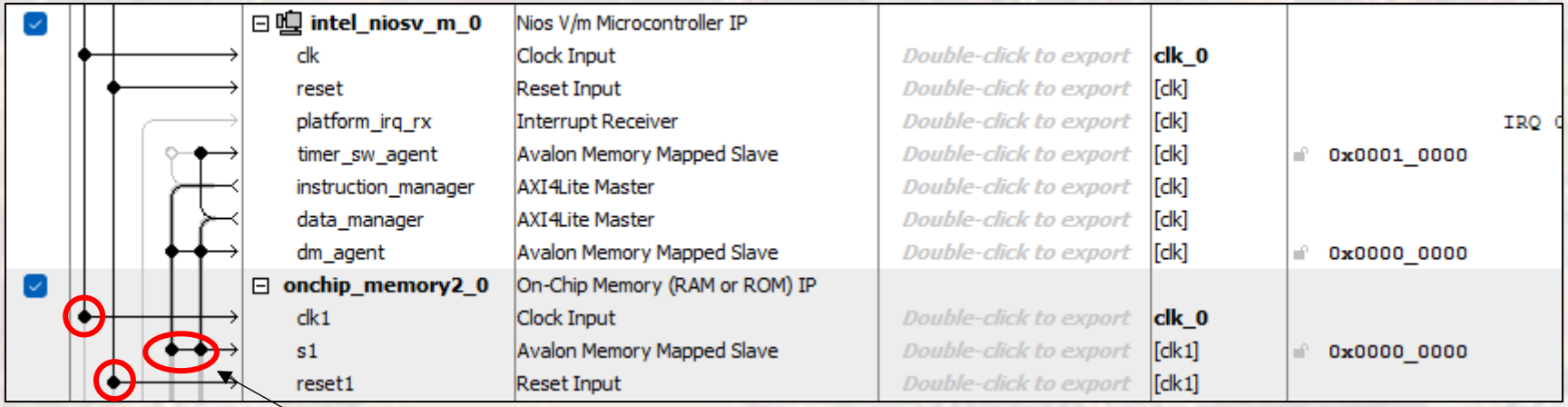
- Connect up basic NIOS system
  - NIOS-V Inputs
    - Clock and Reset



 © tj

NIOS-V – Hardware Setup

- Connect up basic NIOS system
  - On-chip Memory
    - clk, reset, instruction manager, data manager

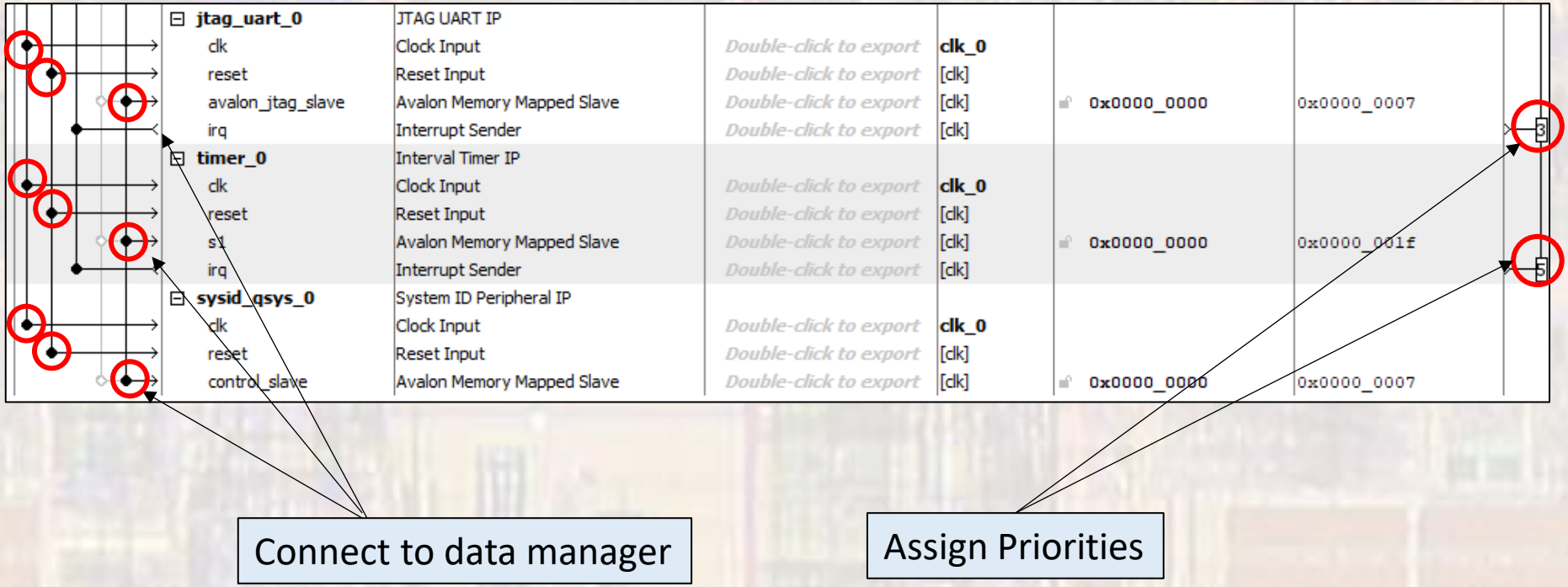


Connect to data and instruction masters

 © tj

NIOS-V – Hardware Setup

- Connect up basic NIOS system
  - JTAG, Timer, SysID
  - 16 interrupt priority values 1-16



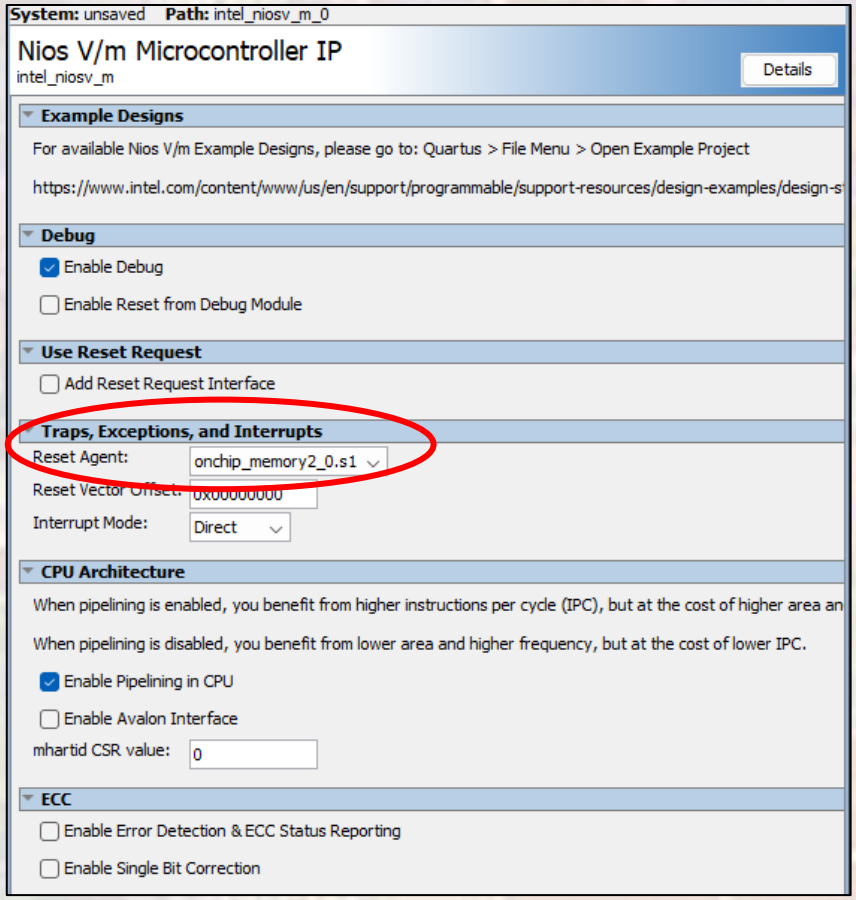
Connect to data manager

Assign Priorities

 © tj

NIOS-V – Hardware Setup

- Connect up basic NIOS system
  - Assign the NIOS-V Reset and Exception vectors
    - Open the NIOS Processor
    - Select Traps, Exceptions, …
    - Select on-chip memory for Reset Agent

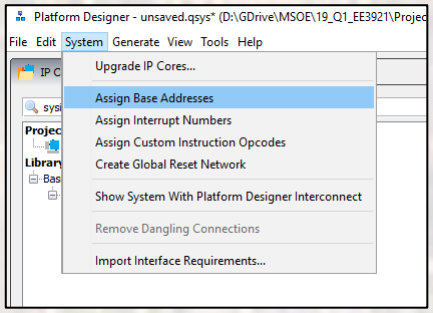


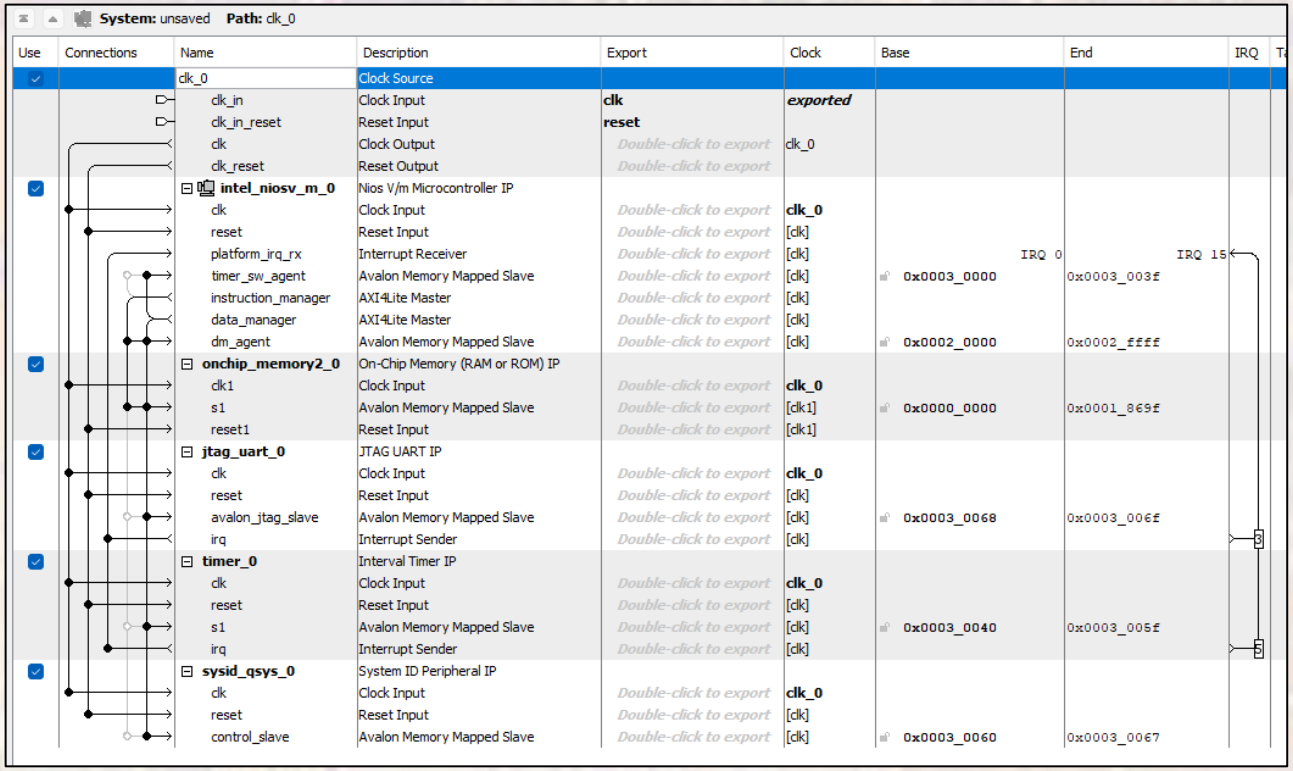
 © tj

NIOS-V – Hardware Setup

- Complete Basic System
  - Assign base addresses
  - System → Assign Base Addresses

Assign Base Addresses





NIOS-V – Hardware Setup

- Create Basic System
  - Check for errors
  - 3 warnings, otherwise all green

These 3 warnings are unavoidable

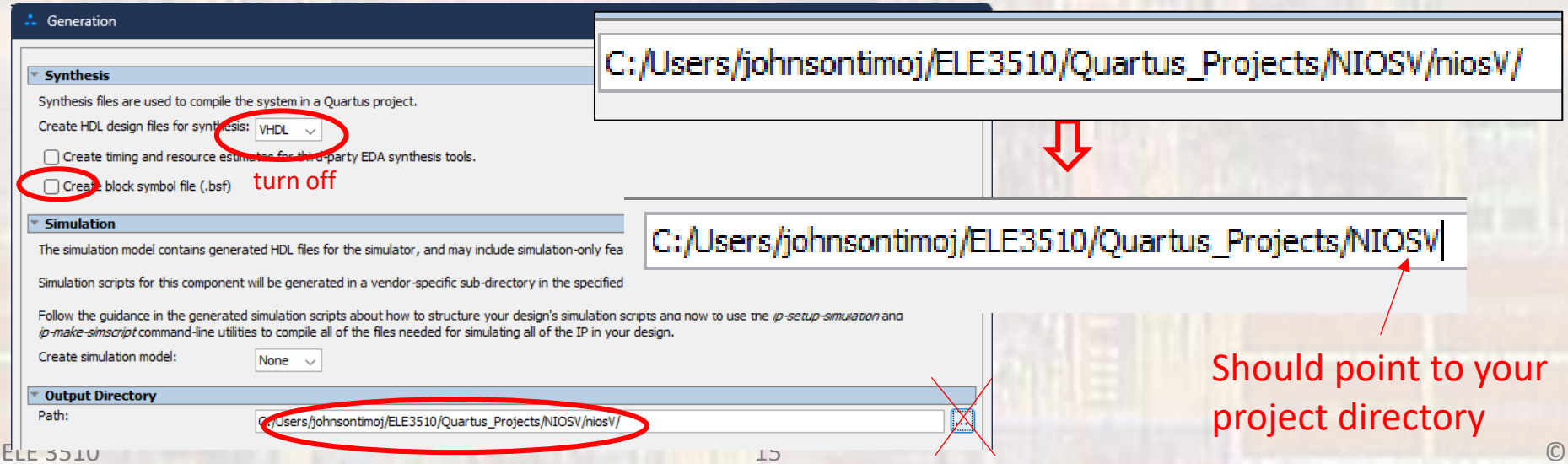
⊟ ⚠	3 Warnings	
⚠	**unsaved.intel_niosv_m_0**	Properties (associatedClock) have been set on interface **reset** - in composed mode these are ignored
⚠	**unsaved.intel_niosv_m_0**	Properties (associatedClock,associatedReset) have been set on interface **instruction_manager** - in composed mode these are ignored
⚠	**unsaved.intel_niosv_m_0**	Properties (associatedClock,associatedReset) have been set on interface **data_manager** - in composed mode these are ignored
⊟ ⓘ	3 Info Messages	
ⓘ	**unsaved.jtag_uart_0**	JTAG UART IP input clock need to be at least double (2x) the operating frequency of JTAG TCK on board
ⓘ	**unsaved.sysid_qsys_0**	System ID is not assigned automatically. Edit the System ID parameter to provide a unique ID
ⓘ	**unsaved.sysid_qsys_0**	Time stamp will be automatically updated when this component is generated.

© tj

NIOS-V – Hardware Setup

- Create Basic System
  - Save the Platform Designer system
    - File → save as → an appropriate name
  - Generate the Platform Designer system
    - Generate → Generate HDL
    - The first time you generate you must delete the last directory in the path – don't use the '…'

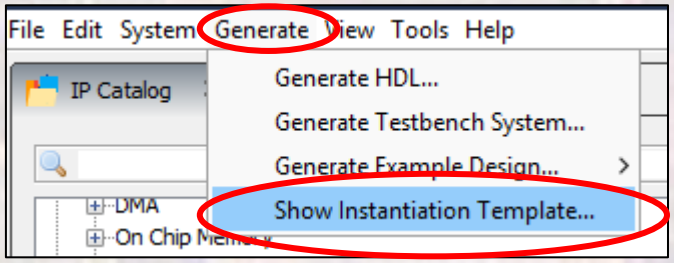
      be sure to delete the backslash before the name

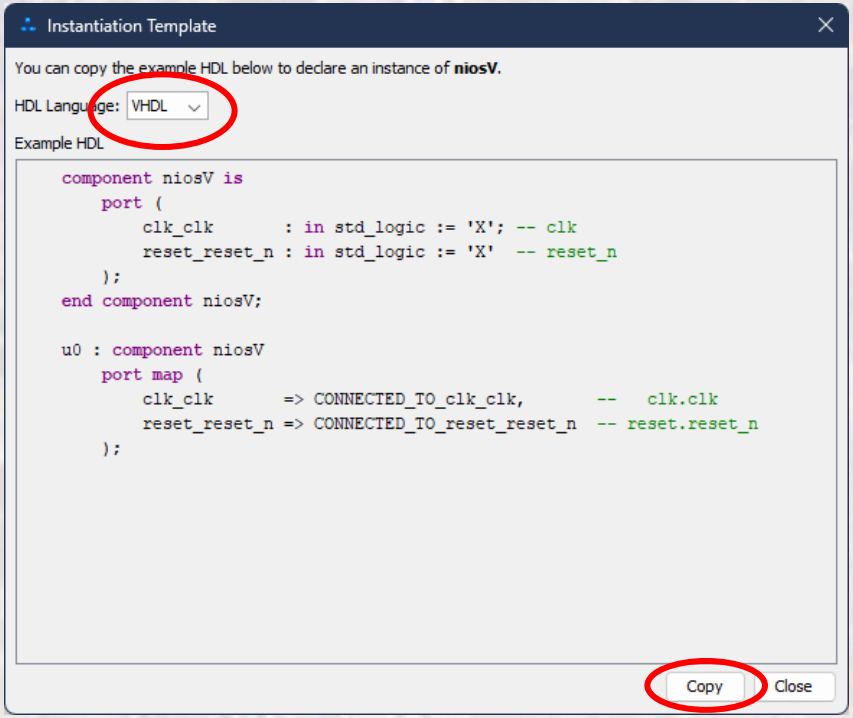


© tj

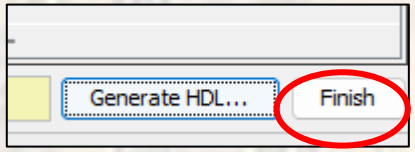
NIOS-V – Hardware Setup

- Create Basic System
  - Create a component instantiation template
    - Generate → Show Instantiation Template
    - Copy the text into a file





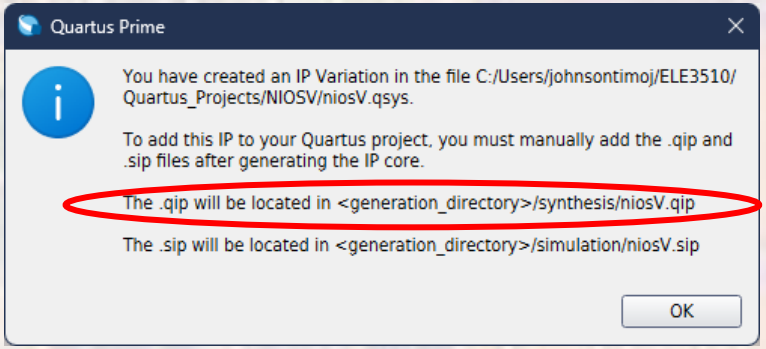
- Finish

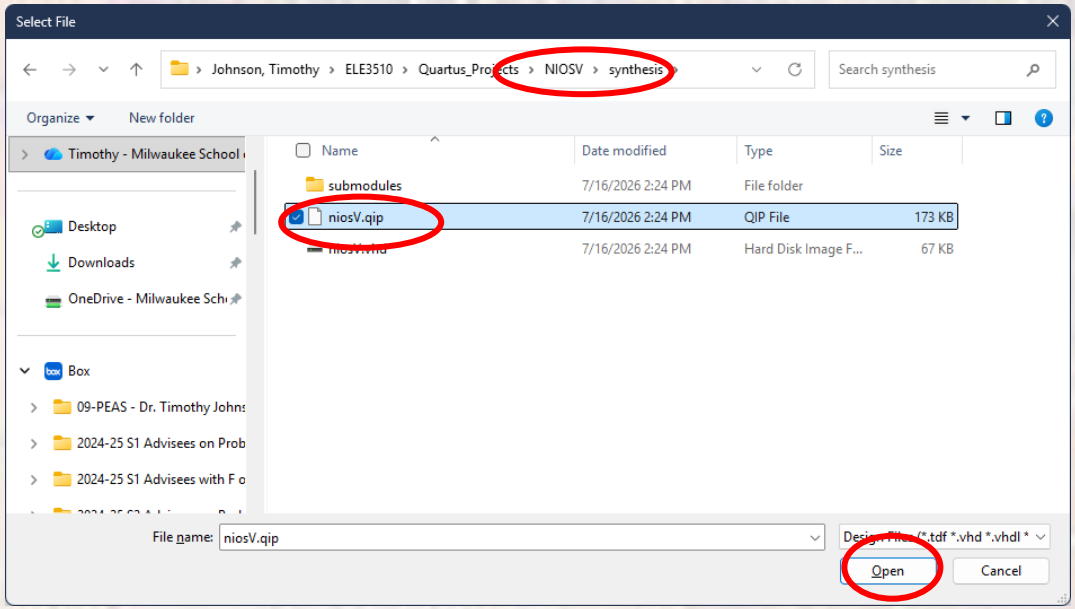


 © tj

NIOS-V – Hardware Setup

- Create Basic System
  - Add the .qip file to the project
    - Project → Add / Remove Files → …
    - Select /synthesis / file name





 © tj

NIOS-V – Hardware Setup

- Create DE10 Design
  - Instantiate into a VHDL file

```
----------------------
--
-- niosV_de10.vhdl
--
-- created 7/16/26
-- by tj
----------------------
--
-- NIOS basic example
--
----------------------
library ieee;
use ieee.std_logic_1164.all;

entity niosV_de10 is
   port(
      CLOCK_50:   in std_logic
   );
end entity;
```

Instantiation template component

```
architecture hardware of niosV_de10 is

   component niosV is
        port (
            clk_clk                         : in  std_logic
            reset_reset_n                   : in  std_logic
        );
   end component;

begin

   u0 : component niosV
        port map (
            clk_clk                         => CLOCK_50,
            reset_reset_n                   => '1'
        );

end architecture;
```

Instantiation template instance mapped to DE10 qsf pin aliases

© tj

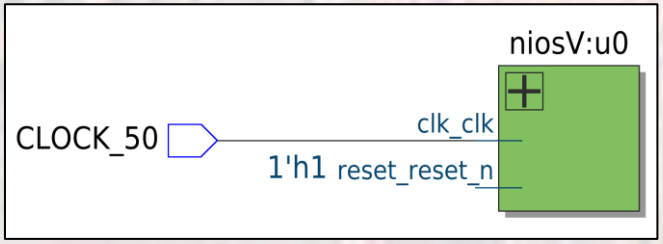
NIOS-V – Hardware Setup

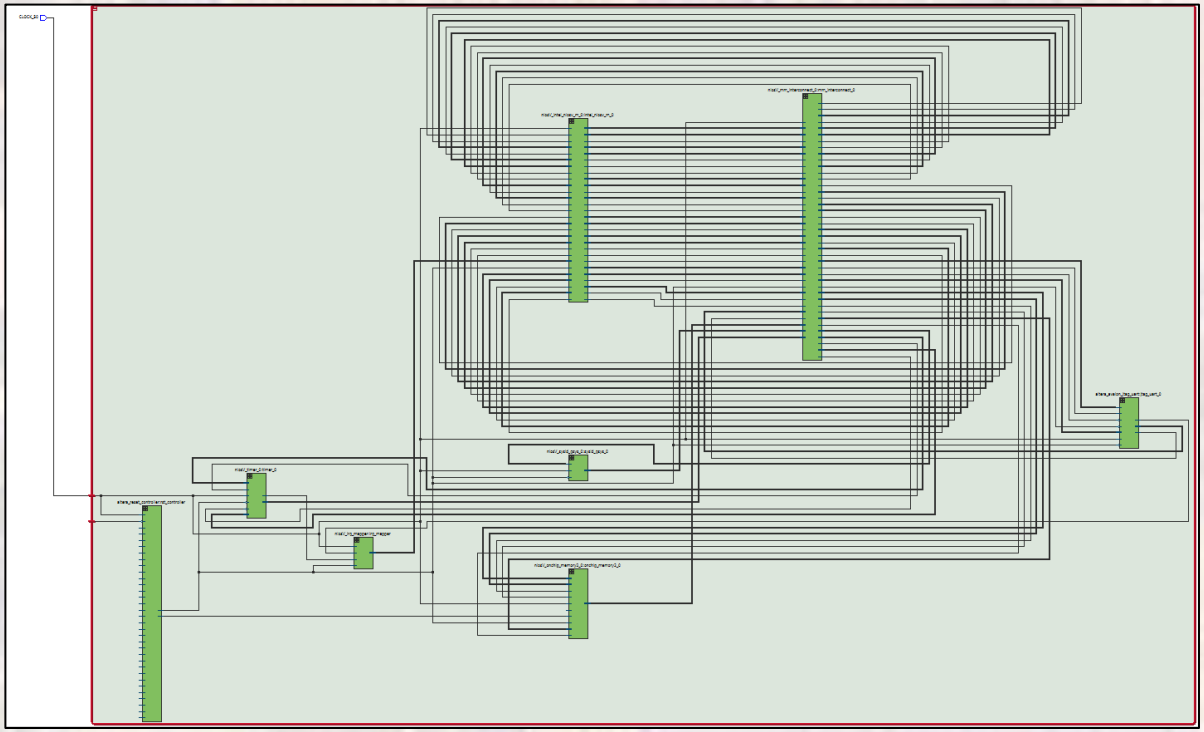
- Create DE10 Design
  - Prepare to synthesize
    - If you did not do these when you created the project be sure to do them now
      - assignments → device → device and Pin options → Configuration Mode → Single Uncompressed with memory initialization
      - Import the pin aliases (qsf file)
      - Setup the SDC file for timing analysis
  - Be sure to set your top level entity to the DE10 file
  - Start Compilation

© tj

NIOS-V – Hardware Setup

- Create DE10 Design

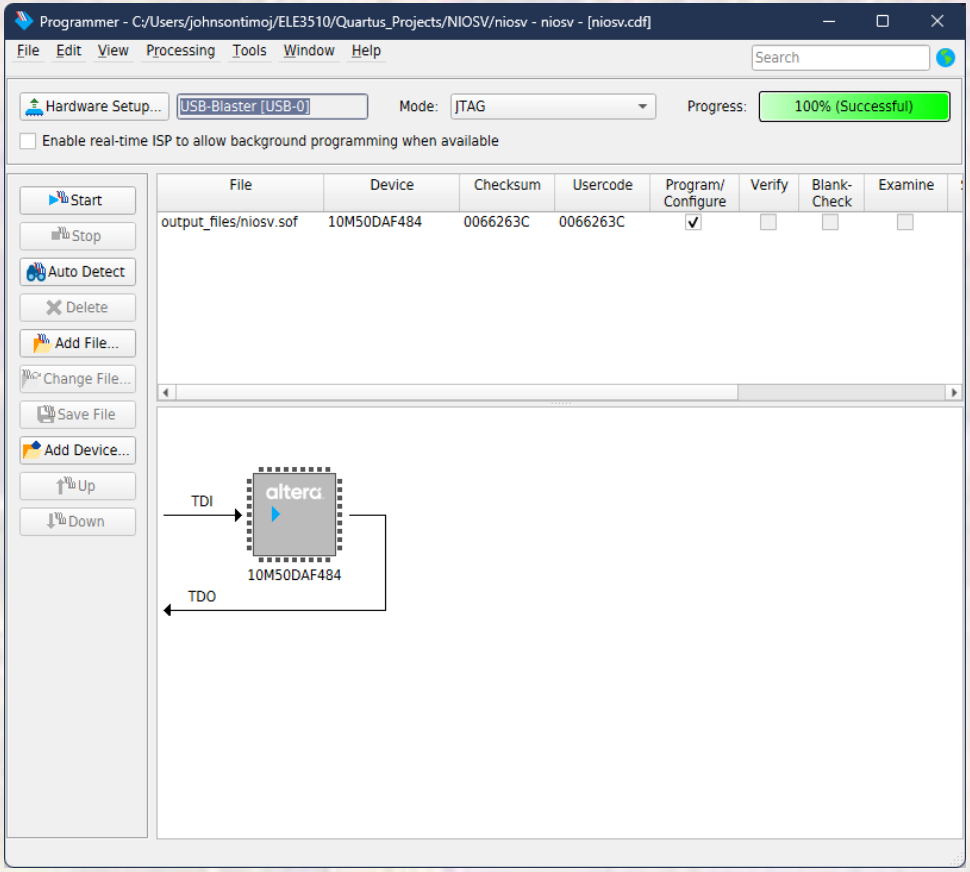




© tj

NIOS-V – Hardware Setup

- Create DE10 Design
  - Complete the HW setup
    - Download the HW project onto the board



 © tj